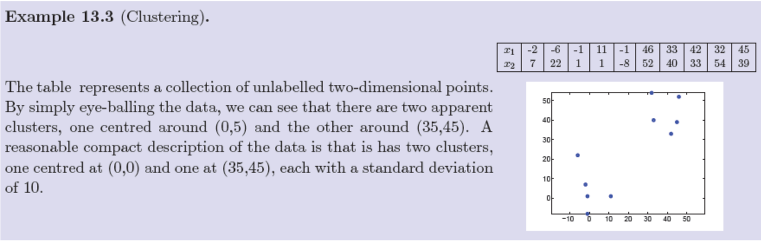
Clustering

- Introduction
- Mixture Densities
- Expectation-Maximization Algorithm
- K-Means Clustering
- Fuzzy Clustering
- Spectral Clustering
- Hierarchical Clustering
- Cluster Validity

Introduction (1)

- Clustering
  - To group the data $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ into a number of clusters
    - Even when there is no nature grouping in the data



Example 13.3 (Clustering).

x_1	-2	-6	-1	11	-1	46	33	42	32	45
x_2	7	22	1	1	-8	52	40	33	54	39

The table represents a collection of unlabelled two-dimensional points. By simply eye-balling the data, we can see that there are two apparent clusters, one centred around (0,5) and the other around (35,45). A reasonable compact description of the data is that is has two clusters, one centred at (0,0) and one at (35,45), each with a standard deviation of 10.

[Barber, 2013]

Introduction (2)

- Basic steps
  - Feature selection
  - Proximity measure
    - To quantify how similar or dissimilar two feature vectors are
  - Clustering criterion
    - Maybe expressed via a cost function or some other rules
  - Clustering algorithms
  - Validation of the results
    - To verify the correctness of the results
  - Interpretation of the results

Evaluation of the final clustering is influenced by domain / expert knowledge

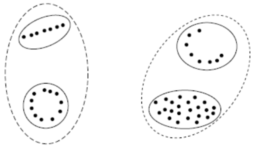


Fig. 11.2 [Theodoridis 09]

Introduction (3)

- Definition of clustering
  - Hard clustering
    - Given the data set $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$
    - The partition of X into K clusters $C_1, \ldots, C_K$ by extremizing a criterion function, so that
      - $C_i \neq \emptyset, i = 1, \ldots, K$
      - $\cup_{i=1}^{K} C_i = X$
      - $C_i \cap C_j = \emptyset,\ i \neq j$
      - The vectors contained in a cluster C_i are more similar to each other and less similar to the feature vectors of the other clusters

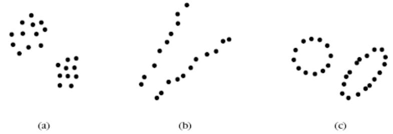


FIGURE 11.3

(a) Compact clusters. (b) Elongated clusters. (c) Spherical and ellipsoidal clusters.

[Theodoridis et. al.]

Introduction (4)

- Definition of clustering (cont.)
  - Fuzzy clustering
    - Each vector belongs to more than one cluster up to some degree, quantified by the K membership functions
    - $u_j : X \to [0,1],\ j = 1,2,\dots,K$
      - $\sum_{j=1}^{K} u_j(\mathbf{x}_i) = 1,\ i = 1,\dots,N$
      - $0 < \sum_{i=1}^{N} u_j(\mathbf{x}_i) < N,\ j = 1,2,\dots,K$
      - Values close to 1
        - » High grade of membership in the corresponding cluster
    - Hard clustering can be seen as a special case if we define the membership function to take values in $\{0,1\}$

Introduction (5)

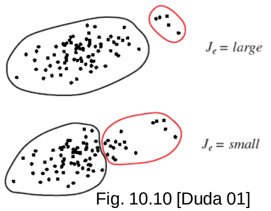


Fig. 10.10 [Duda 01]

- Clustering criterion
  - Sum-of-squared error criterion
    - Minimum variance partition
      - To minimize
        - » $J_e = \sum_{j=1}^{K} \sum_{\mathbf{x} \in C_j} \left\| \mathbf{x} - \boldsymbol{\mu}_j \right\|^2$
        - » where $\boldsymbol{\mu}_j = \frac{1}{|C_j|} \sum_{\mathbf{x} \in C_j} \mathbf{x}$
      - Appropriate when the clusters form compact clouds that are rather well-separated from one another
    - In fuzzy clustering scheme
      - To minimize
        - » $J = \sum_{j=1}^{K} \sum_{i=1}^{N} u_j^q(\mathbf{x}_i) \left\| \mathbf{x}_i - \boldsymbol{\mu}_j \right\|^2$
      - Subject to
        - » $\sum_{j=1}^{K} u_j(\mathbf{x}_i) = 1,\ i = 1,\dots,N$
        - » $u_j(\mathbf{x}_i) \in [0,1]$

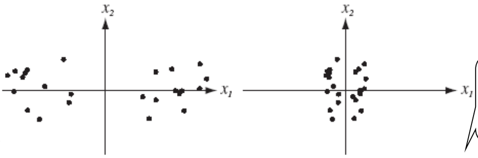
Introduction (6)

- Clustering criterion (cont.)
  - Scatter criteria
    - Minimization of
      - $tr(\mathbf{S}_W) = \sum_{j=1}^{K} P_j tr(\mathbf{S}_j) = \frac{1}{N}\sum_{j=1}^{K}\sum_{\mathbf{x}\in C_j}\left\|\mathbf{x}-\boldsymbol{\mu}_j\right\|^2 \propto J_e$
      - $J_d = |\mathbf{S}_W|$ or $\frac{|\mathbf{S}_W|}{|\mathbf{S}_M|} = \left|\mathbf{S}_M^{-1}\mathbf{S}_W\right|$
      - $J_f = tr(\mathbf{S}_M^{-1}\mathbf{S}_W)$
    - Maximization of
      - $tr(\mathbf{S}_B) = \sum_{j=1}^{K} P_j \left\|\boldsymbol{\mu}_j - \boldsymbol{\mu}\right\|^2$
      - $tr(\mathbf{S}_W^{-1}\mathbf{S}_B)$
  - Graph cut
    - Minimization of
      - $cut(C_1,\ldots,C_K) = \frac{1}{2}\sum_{j=1}^{K} W\left(C_j,\bar{C}_j\right)$
      - $Ncut(C_1,\ldots,C_K) = \frac{1}{2}\sum_{j=1}^{K} cut\left(C_j,\bar{C}_j\right)/vol\left(C_j\right)$

$tr(\mathbf{S}_M) = tr(\mathbf{S}_W) + tr(\mathbf{S}_B)$
$\Rightarrow \min tr(\mathbf{S}_W) = \max tr(\mathbf{S}_B)$

Introduction (7)

- Normalization
  - Normalizing the data prior to clustering
    - e.g. normalizing into zero mean and unit variance
    - e.g. rotating the axes of the feature space by PCA
  - Normalization may reduce the separation between well-separated data



Well-separated clusters

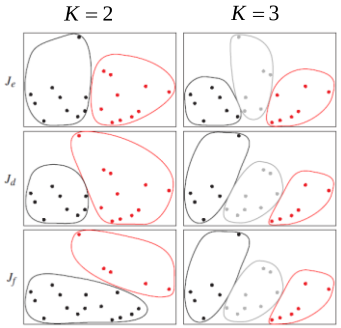
Normalization by scaling for unit variance reduces the separation

Fig. 10.9 [Duda 01]

FIGURE 10.9. If the data fall into well-separated clusters (left), normalization by scaling for unit variance for the full data may reduce the separation, and hence be undesirable (right). Such a normalization may in fact be appropriate if the full data set arises from a single fundamental process (with noise), but inappropriate if there are several different processes, as shown here. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Introduction (8)

- Example (p.547, [Duda 01])
  - Original data
    - Do not exhibit obvious clusters
  - Using different criteria and #clusters



J_e favors clusters of roughly equal number of points

The difference between the cluster criteria become less pronounced for large number of clusters

Mixture Densities (1)

- Mixture density
  - Given N unlabeled samples $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ drawn independently from a mixture of K models
    - $p(\mathbf{x}) = \sum_{j=1}^{K} p(\mathbf{x}|C_j) P(C_j)$
      - $P(C_j)$: the mixing proportions for each model
        - » $\sum_{j=1}^{K} P(C_j) = 1$
      - $p(\mathbf{x}|C_j)$: The model density
        - » $\int_{\mathbf{x}} p(\mathbf{x}|C_j) d\mathbf{x} = 1$
  - Parametric form
    - Let $p(\mathbf{x}|C_j) \equiv p(\mathbf{x}|C_j, \boldsymbol{\theta}_j)$
    - $p(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{j=1}^{K} p(\mathbf{x}|C_j, \boldsymbol{\theta}_j) P(C_j)$
    - The unknown parameter $\Theta = [\boldsymbol{\theta}, \mathbf{P}]$
      - $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K]$ and $\mathbf{P} = [P(C_1), \dots, P(C_K)]$

Mixture Densities (2)

- Clustering
  - Soft clustering
    - $P(C_i|\mathbf{x}) = \frac{p(\mathbf{x}|C_i,\boldsymbol{\theta}_i)P(C_i)}{\sum_{j=1}^{K} p(\mathbf{x}|C_j,\boldsymbol{\theta}_j)P(C_j)}$
  - Hard clustering
    - $\mathbf{x} \to C_j$
    - $j = \underset{i}{\operatorname{argmax}} \log P(C_i|\mathbf{x}) = \underset{i}{\operatorname{argmax}} \log p(\mathbf{x}|C_i,\boldsymbol{\theta}_i) + \log P(C_i)$

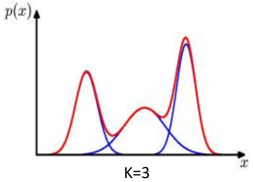


Fig. 2.22 [Bishop 06]

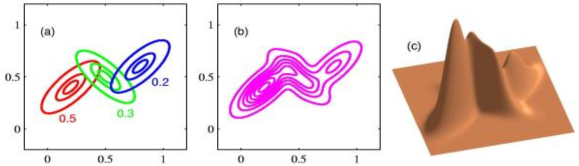


Fig. 2.23 [Bishop 06]

Mixture Densities (3)

- Why not using ML?
  - Given $X = \{\mathbf{x}_1,\mathbf{x}_2, \dots, \mathbf{x}_N\}$ drawn independently from
    - $p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{j=1}^{K} p\big(\mathbf{x}|C_j,\boldsymbol{\theta}_j\big)P(C_j)$
      - Where $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K]$ is fixed but unknown
  - ML estimate of $\boldsymbol{\theta}$
    - $\hat{\boldsymbol{\theta}}_{ML} = \underset{\boldsymbol{\theta}}{argmax}\, p(X|\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{argmax} \prod_{k=1}^{N} p(\mathbf{x}_k|\boldsymbol{\theta})$

      $= \underset{\boldsymbol{\theta}}{argmax} \prod_{k=1}^{N} \sum_{j=1}^{K} p\big(\mathbf{x}_k|C_j,\boldsymbol{\theta}_j\big)P(C_j)$
  - The log-likelihood
    - $L(\boldsymbol{\theta}) = ln\, p(X|\boldsymbol{\theta}) = \sum_{k=1}^{N} \ln p(\mathbf{x}_k|\boldsymbol{\theta}) = \sum_{k=1}^{N} \ln\big[\sum_{j=1}^{K} p\big(\mathbf{x}_k|C_j,\boldsymbol{\theta}_j\big)P(C_j)\big]$
    - $\hat{\boldsymbol{\theta}}_{ML} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, L(\boldsymbol{\theta})$

Mixture Densities (4)

- Why not using ML? - Gaussian mixtures case
  - Assume the model densities are multivariate normal
    - $p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{j=1}^{K} p\left(\mathbf{x}|C_j, \boldsymbol{\theta}_j\right)P(C_j)$
      - $p\left(\mathbf{x}|C_j, \boldsymbol{\theta}_j\right) \sim N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$
  - Solving mean vectors
    - $L(\boldsymbol{\theta}) = ln\, p(X|\boldsymbol{\theta}) = \sum_{k=1}^{N} \ln p(\mathbf{x}_k|\boldsymbol{\theta}) = \sum_{k=1}^{N} \ln\left[\sum_{j=1}^{K} p\left(\mathbf{x}_k|C_j, \boldsymbol{\theta}_j\right)P(C_j)\right]$
    - The ML estimate must satisfy
      - $\nabla_{\boldsymbol{\theta}} L \equiv \frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{k=1}^{N} \nabla_{\boldsymbol{\theta}_i} \ln p(\mathbf{x}_k|\boldsymbol{\theta})$

$$= \sum_{k=1}^{N} \frac{1}{p(\mathbf{x}_k|\boldsymbol{\theta})} \nabla_{\boldsymbol{\theta}_i}\left[\sum_{j=1}^{K} p\left(\mathbf{x}_k|C_j, \boldsymbol{\theta}_j\right)P(C_j)\right] = 0$$

$$\Rightarrow \sum_{k=1}^{N} \frac{1}{p(\mathbf{x}_k|\boldsymbol{\theta})} P(C_i)\nabla_{\boldsymbol{\theta}_i}[p(\mathbf{x}_k|C_i, \boldsymbol{\theta}_i)] = \cdots$$

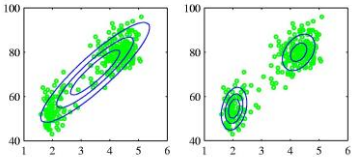
$$= \sum_{k=1}^{N} P(C_i|\mathbf{x}_k, \boldsymbol{\theta})\nabla_{\boldsymbol{\theta}_i}\ln[p(\mathbf{x}_k|C_i, \boldsymbol{\theta}_i)] = 0$$

$$\Rightarrow \sum_{k=1}^{N} P\left(C_i|\mathbf{x}_k, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}\right)\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_i) = 0$$

$$\Rightarrow \sum_{k=1}^{N} P\left(C_i|\mathbf{x}_k, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}\right)\mathbf{x}_k = \left(\sum_{k=1}^{N} P\left(C_i|\mathbf{x}_k, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}\right)\right)\hat{\boldsymbol{\mu}}_i$$

Mixture Models (5)

- Why not using ML?- Gaussian mixtures case (cont.)
  - The ML estimate is
    - $\hat{P}(C_i) = \frac{1}{N}\sum_{k=1}^{N} \hat{P}\left(C_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}}\right)$
    - $\hat{\boldsymbol{\mu}}_i = \frac{\sum_{k=1}^{N} \hat{P}\left(C_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}}\right)\mathbf{x}_k}{\sum_{k=1}^{N} \hat{P}\left(C_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}}\right)}$
    - $\hat{\boldsymbol{\Sigma}}_i = \frac{\sum_{k=1}^{N} \hat{P}\left(C_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}}\right)(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_i)(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_i)^T}{\sum_{k=1}^{N} \hat{P}\left(C_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}}\right)}$



ML estimated Single Gaussian

Mixture of two Gaussians (J=2)

Fig. 2.21 [Bishop 06]

  - However, the equation does not give the solution explicitly
    - $\hat{P}\left(C_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}}\right) = \frac{p\left(\mathbf{x}_k|C_i, \hat{\boldsymbol{\theta}}_i\right)\hat{P}(C_i)}{\sum_{j=1}^{K} p\left(\mathbf{x}_k|C_j, \hat{\boldsymbol{\theta}}_i\right)\hat{P}(C_j)}$

Need to solve a set of nonlinear equations

      - where $p\left(\mathbf{x}|C_i, \hat{\boldsymbol{\theta}}_i\right) \sim N(\hat{\boldsymbol{\mu}}_{i_j}, \hat{\boldsymbol{\Sigma}}_i)$

unknown

Expectation-Maximization (1)

- EM algorithm
  - To find the ML solutions for models having hidden variables
    - An iterative algorithm which alternates between
      - E step:
        - » Inferring the missing values given the parameters
      - M step:
        - » Optimizing the parameters given the filled-in data
  - Given the **incomplete** data set $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$
  - Let the **complete** data set be $Y = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$
    - $\mathbf{y}_k = [\mathbf{x}_k, \mathbf{z}_k]$
      - Assume $\mathbf{y}_k$ are taken from $p(\mathbf{y}|\boldsymbol{\theta})$
  - The complete data log likelihood is
    - $ln\, p(Y|\boldsymbol{\theta}) = \sum_{k=1}^{N} \ln p(\mathbf{y}_k|\boldsymbol{\theta}) = \sum_{k=1}^{N} \ln p(\mathbf{x}_k, \mathbf{z}_k|\boldsymbol{\theta})$
      - Can NOT be computed
        - » Because $\mathbf{z}_k$ is unknown

Expectation-Maximization (2)

- EM algorithm (cont.)
  - The expected complete data log likelihood
    - $Q(\boldsymbol{\theta}; \boldsymbol{\theta}(t)) = E_Z\{\ln p(Y|\boldsymbol{\theta})\, | X, \boldsymbol{\theta}(t)\}$

      $= \sum_Z P(Z|X, \boldsymbol{\theta}(t)) \ln P(Y|\boldsymbol{\theta})$ (or $\int_Z P(Z|X, \boldsymbol{\theta}(t)) \ln P(Y|\boldsymbol{\theta})\, dZ$)
    - $Q(\boldsymbol{\theta}; \boldsymbol{\theta}(t))$: the auxiliary function of $\boldsymbol{\theta}$ with $\boldsymbol{\theta}(t)$ assumed fixed
      - $\boldsymbol{\theta}(t)$ is the current (best) estimate
    - The expectation is taken over the unobserved data wrt the observed samples X and the current estimate of $\boldsymbol{\theta}(t)$
  - E step
    - To evaluate $Q(\boldsymbol{\theta}; \boldsymbol{\theta}(t))$ or the terms $P(Z|X, \boldsymbol{\theta}(t))$
  - M step
    - To optimize the Q function with respect to $\boldsymbol{\theta}$
      - $\boldsymbol{\theta}(t+1) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, Q(\boldsymbol{\theta}; \boldsymbol{\theta}(t))$

M step in the MAP estimation

$\boldsymbol{\theta}(t+1) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, Q(\boldsymbol{\theta}; \boldsymbol{\theta}(t)) + \ln p(\boldsymbol{\theta})$

Expectation-Maximization (3)

- EM algorithm
  - Initial estimate $\boldsymbol{\theta}(0)$
  - Iteration
    - E step
      - Evaluate the posterior of Z
        - » $P(Z|X, \boldsymbol{\theta}(t))$
    - M step
      - $\frac{\partial}{\partial \boldsymbol{\theta}} Q(\boldsymbol{\theta}; \boldsymbol{\theta}(t)) = 0$
      - $\boldsymbol{\theta}(t+1) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, Q(\boldsymbol{\theta}; \boldsymbol{\theta}(t))$
    - Until
      - $Q(\boldsymbol{\theta}(t+1); \boldsymbol{\theta}(t)) - Q(\boldsymbol{\theta}(t); \boldsymbol{\theta}\,(t-1)) \leq T$
      - Or $\|\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)\| \leq \boldsymbol{\varepsilon}$

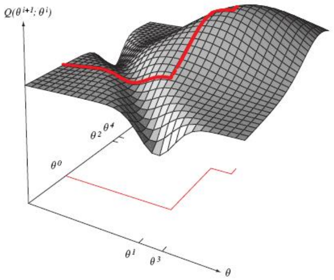


Fig. 3.7 [Duda 01]

Expectation-Maximization (4)

- EM algorithm for Gaussian mixtures $\boldsymbol{\theta}_j = \{P_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}$
  - E-step
    - Evaluate the posterior of j
      - $P(j|\mathbf{x}_k, \boldsymbol{\theta}(t)) = \frac{p(\mathbf{x}_k|j,\boldsymbol{\theta}(t))P(j)}{p(\mathbf{x}_k,\boldsymbol{\theta}(t))} = \frac{p(\mathbf{x}_k|j,\boldsymbol{\theta}(t))P(j)}{\sum_{i=1}^{K} p(\mathbf{x}_k|i,\boldsymbol{\theta}(t))P(i)}$

$$= \frac{|\boldsymbol{\Sigma}_j(t)|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu}_j(t))^{\boldsymbol{T}} \boldsymbol{\Sigma}_j^{-\mathbf{1}}(t)(\mathbf{x}_k - \boldsymbol{\mu}_j(t))\right] P(j)}{\sum_{i=1}^{K} |\boldsymbol{\Sigma}_i(t)|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu}_i(t))^{\boldsymbol{T}} \boldsymbol{\Sigma}_i^{-\mathbf{1}}(t)(\mathbf{x}_k - \boldsymbol{\mu}_i(t))\right] P(i)}$$

  - M-step
    - Maximize Q (constraint $\sum_{i=1}^{K} P(i) = 1$)
      - $\boldsymbol{\mu}_j(t+1) = \frac{\sum_{k=1}^{N} P(j|\mathbf{x}_k,\boldsymbol{\theta}(t))\mathbf{x}_k}{\sum_{k=1}^{N} P(j|\mathbf{x}_k,\boldsymbol{\theta}(t))}$
      - $\boldsymbol{\Sigma}_j(t+1) = \frac{\sum_{k=1}^{N} P(j|\mathbf{x}_k,\boldsymbol{\theta}(t))(\mathbf{x}_k - \boldsymbol{\mu}_j)(\mathbf{x}_k - \boldsymbol{\mu}_j)^T}{\sum_{k=1}^{N} P(j|\mathbf{x}_k,\boldsymbol{\theta}(t))}$
      - $P_j(t+1) = \frac{1}{N} \sum_{k=1}^{N} P(j|\mathbf{x}_k, \boldsymbol{\theta}(t))$

Expectation-Maximization (5)

- Example (p.48, [Theodoridis 09])
  - $N = 100$ samples are drawn from a mixture of 2 Gaussian models
    - $p(\mathbf{x}) = P \times g(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1-P) \times g(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$
      - $P = 0.8$
      - $\boldsymbol{\mu}_1 = \begin{bmatrix}1\\1\end{bmatrix}$ $\boldsymbol{\mu}_2 = \begin{bmatrix}2\\2\end{bmatrix}$ $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \begin{bmatrix}0.1 & 0\\0 & 0.11\end{bmatrix}$
  - The unknown parameter vector
    - $\boldsymbol{\theta} = \{P, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2\}$
    - Initial values
      - $P = 0.5$
      - $\boldsymbol{\mu}_1 = \begin{bmatrix}1.37\\1.2\end{bmatrix}, \boldsymbol{\mu}_2 = \begin{bmatrix}1.81\\1.62\end{bmatrix} \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \begin{bmatrix}0.44 & 0\\0 & 0.44\end{bmatrix}$
    - After convergence
      - $P = 0.844$
      - $\boldsymbol{\mu}_1 = \begin{bmatrix}1.05\\1.03\end{bmatrix}, \boldsymbol{\mu}_2 = \begin{bmatrix}1.9\\2.08\end{bmatrix} \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \begin{bmatrix}0.1 & 0\\0 & 0.06\end{bmatrix}$

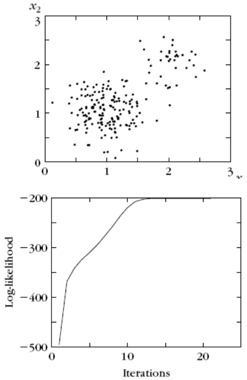


Fig. 2.17 [Theodoridis 09]

Expectation-Maximization (6)

- Example (p. 437, [Bishop 06])

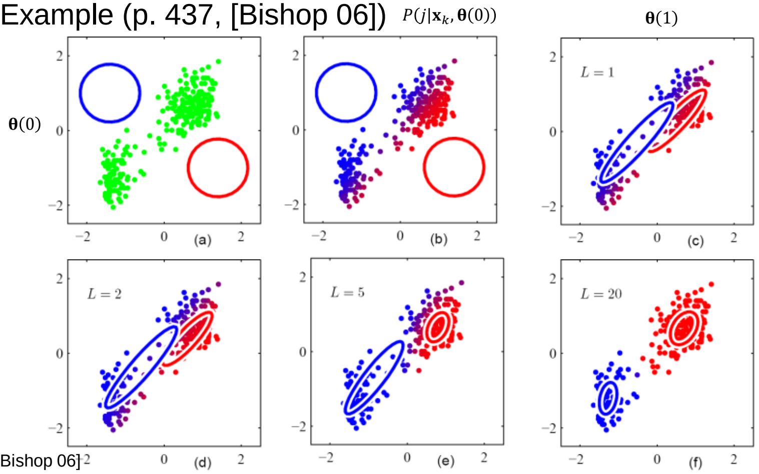


Fig. 9.8 [Bishop 06]

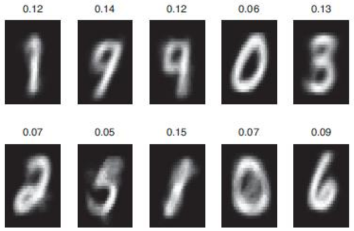


Figure 11.5 We fit a mixture of 10 Bernoullis to the binarized MNIST digit data. We show the MLE for the corresponding cluster means, μ_k. The numbers on top of each image represent the mixing weights $\hat{\pi}_k$. No labels were used when training the model. Figure generated by `mixBerMnistEM`.

[Murphy]

The clustering results are not good

- Multiple clusters for some digits (e.g., 9 and 0)
- No clusters for others (e.g., 4 and 7)

Possible reasons

- Each pixel is treated independently and no visual characteristics of a digit are captured
- The number of clusters may be > 10
- The algorithm may stuck in a local optimum

Expectation-Maximization (7)

- Example 14.1.a (p.706, [Theodoridis 09])
  - 3x100 vectors generated from three 2-D normal distributions with
    - $\boldsymbol{\mu}_1 = \begin{bmatrix}1\\1\end{bmatrix}$ $\boldsymbol{\mu}_2 = \begin{bmatrix}3.5\\3.5\end{bmatrix}$ $\boldsymbol{\mu}_3 = \begin{bmatrix}6\\1\end{bmatrix}$
    - $\boldsymbol{\Sigma}_1 = \begin{bmatrix}1 & -0.3\\-0.3 & 1\end{bmatrix}$ $\boldsymbol{\Sigma}_2 = \begin{bmatrix}1 & 0.3\\0.3 & 1\end{bmatrix}$ $\boldsymbol{\Sigma}_3 = \begin{bmatrix}1 & 0.7\\0.7 & 1\end{bmatrix}$

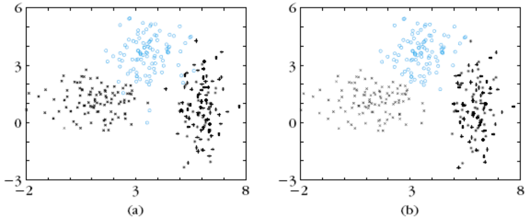


Fig. 14.3 [Theodoridis 09]

Confusion matrix

	C1	C2	C3
1st distribution	99	0	1
2nd distribution	0	100	0
3rd distribution	3	4	93

Expectation-Maximization (8)

- Example 14.1.b (p.708, [Theodoridis 09])
  - 3x100 vectors generated from three 2-D normal distributions with
    - $\boldsymbol{\mu}_1 = \begin{bmatrix}1\\1\end{bmatrix}$ $\boldsymbol{\mu}_2 = \begin{bmatrix}2\\2\end{bmatrix}$ $\boldsymbol{\mu}_3 = \begin{bmatrix}3\\1\end{bmatrix}$
    - $\boldsymbol{\Sigma}_1 = \begin{bmatrix}1 & -0.3\\-0.3 & 1\end{bmatrix}$ $\boldsymbol{\Sigma}_2 = \begin{bmatrix}1 & 0.3\\0.3 & 1\end{bmatrix}$ $\boldsymbol{\Sigma}_3 = \begin{bmatrix}1 & 0.7\\0.7 & 1\end{bmatrix}$

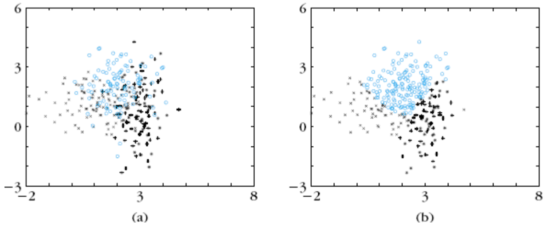


Fig. 14.4 [Theodoridis et. al.]

Confusion matrix

	C1	*C2*	*C3*
1st distribution	85	4	11
2nd distribution	35	56	9
3rd distribution	26	0	74

Expectation-Maximization (9)

- Example 14.2 (p. 708 [Theodoridis 09])
  - The data set consists of 2 intersecting ring-shaped clusters
    - Each cluster consists of 500 points
  - GMM fails to represent the underlying clustering structure

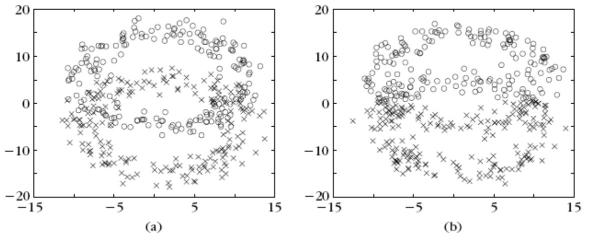


FIGURE 14.5

(a) A data set that consists of ring-shaped intersecting clusters. (b) The results from the application of GMDAS when Gaussian mixtures are used.

K-Means Clustering (1)

- K-means clustering
  - A variant of the EM algorithm for GMMs
    - Not a proper EM on ML
  - Assumptions
    - Equal mixing weights $P(j) = 1/K$
    - An equal spherical covariance matrix for each cluster $\boldsymbol{\Sigma}_j = \sigma^2 \mathbf{I}$
      - Only the cluster centers need to be estimated
    - The posterior probability for GMMs is approximated by the delta function

– $P(j|\mathbf{x}_k, \boldsymbol{\theta}(t)) = \frac{p(\mathbf{x}_k|j,\boldsymbol{\theta}(t))P(j)}{\sum_{i=1}^{K} p(\mathbf{x}_k|i,\boldsymbol{\theta}(t))P(i)} = \frac{|\widehat{\boldsymbol{\Sigma}}_j|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x}_k-\widehat{\boldsymbol{\mu}}_j)^T \widehat{\boldsymbol{\Sigma}}_{\mathbf{j}}^{-1}(\mathbf{x}_k-\widehat{\boldsymbol{\mu}}_j)\right]P(j)}{\sum_{i=1}^{K}|\widehat{\boldsymbol{\Sigma}}_i|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x}_k-\widehat{\boldsymbol{\mu}}_i)^T \widehat{\boldsymbol{\Sigma}}_{\mathbf{i}}^{-1}(\mathbf{x}_k-\widehat{\boldsymbol{\mu}}_i)\right]P(i)}$

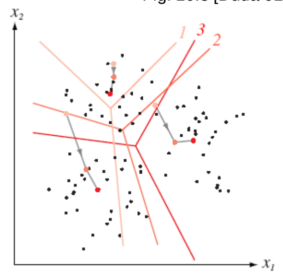
– $P(j|\mathbf{x}_k, \boldsymbol{\theta}(t)) = \begin{cases} 1, & \text{if } j = \underset{i}{\operatorname{argmin}} \|\mathbf{x}_k - \widehat{\boldsymbol{\mu}}_i\|^2 \\ 0, & \text{otherwise} \end{cases}$

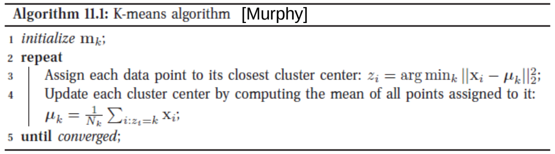
K-Means Clustering (2)

- K-means clustering (cont.)
  - Hard clustering
    - $\mathbf{x} \to C_j$
    - $j = \underset{i}{\operatorname{argmin}} \|\mathbf{x}_k - \widehat{\boldsymbol{\mu}}_i\|^2$
  - The K mean vectors are updated by
    - $\boldsymbol{\mu}_j(t+1) = \frac{\sum_{k=1}^{N} P(j|\mathbf{x}_k,\boldsymbol{\theta}(t))\mathbf{x}_k}{\sum_{k=1}^{N} P(j|\mathbf{x}_k,\boldsymbol{\theta}(t))}$

$$= \frac{1}{number\ of\ samples_{\mathbf{x}_k \to C_j}} \sum_{\mathbf{x}_k \to C_j}^{k} \mathbf{x}_k$$

Fig. 10.3 [Duda 01]





Algorithm 11.1: K-means algorithm [Murphy]

```
initialize $\mathbf{m}_k$;
repeat
    Assign each data point to its closest cluster center: $z_i = \arg\min_k ||\mathbf{x}_i - \mu_k||_2^2$;
    Update each cluster center by computing the mean of all points assigned to it:
    $\mu_k = \frac{1}{N_k} \sum_{i:z_i=k} \mathbf{x}_i$;
until converged;
```

K-Means Clustering (3)

- Example 14.12 (p.742, [Theodoridis 09])
  - (a) Consider the data used in Example 14.1 (a)
    - 3x100 vectors are generated from three 2-D normal distributions with
      - $\boldsymbol{\mu}_1 = \begin{bmatrix}1\\1\end{bmatrix}$ $\boldsymbol{\mu}_2 = \begin{bmatrix}3.5\\3.5\end{bmatrix}$ $\boldsymbol{\mu}_3 = \begin{bmatrix}6\\1\end{bmatrix}$
      - $\boldsymbol{\Sigma}_1 = \begin{bmatrix}1 & -0.3\\-0.3 & 1\end{bmatrix}$ $\boldsymbol{\Sigma}_2 = \begin{bmatrix}1 & 0.3\\0.3 & 1\end{bmatrix}$ $\boldsymbol{\Sigma}_3 = \begin{bmatrix}1 & 0.7\\0.7 & 1\end{bmatrix}$
    - K-means result ($K = 3$)
      - $\hat{\boldsymbol{\mu}}_1 = \begin{bmatrix}1.19\\1.16\end{bmatrix}$ $\hat{\boldsymbol{\mu}}_2 = \begin{bmatrix}3.76\\3.63\end{bmatrix}$ $\hat{\boldsymbol{\mu}}_3 = \begin{bmatrix}5.93\\0.55\end{bmatrix}$

	C1	*C2*	*C3*
1st distribution	94	3	3
2nd distribution	0	100	0
3rd distribution	9	0	91

K-Means Clustering (4)

- Example 14.12 (cont.)
  - (b) Consider two 2-D Gaussian distributions
    - 300 points from the 1st distribution
    - 10 points from the 2nd distribution
      - $\boldsymbol{\mu}_1 = \begin{bmatrix}1\\1\end{bmatrix}$ $\boldsymbol{\mu}_2 = \begin{bmatrix}8\\1\end{bmatrix}$
      - $\boldsymbol{\Sigma}_1 = 1.5\mathbf{I}$ $\boldsymbol{\Sigma}_2 = \mathbf{I}$
    - K-means result ($K = 2$)
      - $\hat{\boldsymbol{\mu}}_1 = \begin{bmatrix}0.54\\0.94\end{bmatrix}$ $\hat{\boldsymbol{\mu}}_2 = \begin{bmatrix}3.53\\0.99\end{bmatrix}$

The algorithm cannot deal accurately with clusters having significantly different sizes

	C1	*C2*
1	239	61
2	0	10

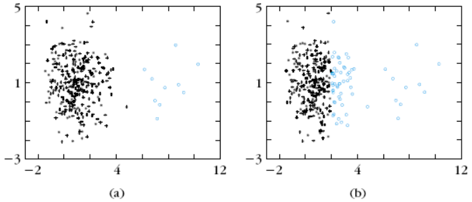


Fig. 14.17 [Theodoridis 09]

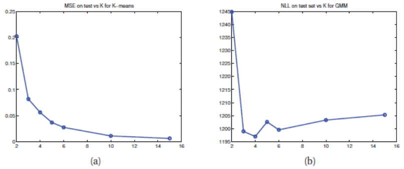


Figure 11.20 Test set performance vs K for data generated from a mixture of 3 Gaussians in 1d (data is shown in Figure 11.21(a)). (a) MSE on test set for K-means. (b) Negative log likelihood on test set for GMM. Figure generated by `kmeansModelSel1d`.

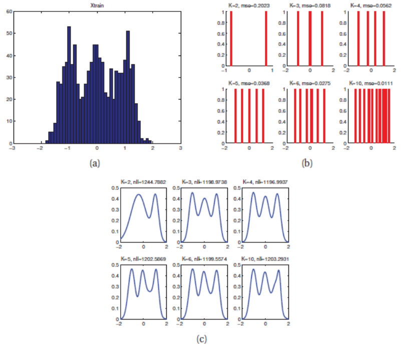


Figure 11.21 Synthetic data generated from a mixture of 3 Gaussians in 1d. (a) Histogram of training data. (Test data looks essentially the same.) (b) Centroids estimated by K-means for $K \in \{2, 3, 4, 5, 6, 10\}$. (c) GMM density model estimated by EM for for the same values of K. Figure generated by `kmeansModelSel1d`.

The reconstruction error decreases with increasing model complexity

[Murphy]

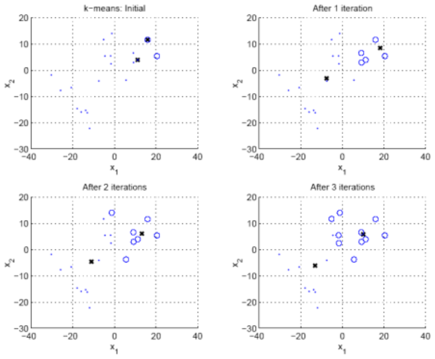
K-Means Clustering (6)

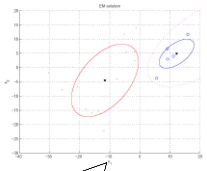
- Initialization
  - The final K means highly depend on the initial
    - Random selection
      - Pick the initial points uniformly at random
    - Farthest point clustering
      - Pick each subsequent point from the remaining points with probability proportional to its squared distance to the closest cluster center
- Choosing K
  - Identify a knee in the curve of reconstruction vs K
  - Incrementally grow GMMs
    - Splitting the cluster with the highest mixture weight into two
    - A cluster is removed if its mixing weight or variance is too small
  - Infinite mixture models
    - Dirichlet process mixture models

K-Means Clustering (7)

- Example
  - Fig. 7.2 [Alpaydin, 2014]
    - Evolution of k-means

Fig 7.4 [Alpaydin]





The fitted Gaussians by EM
(Initialization by one k-means iteration)

Unlike in K-means, EM allows
estimating the covariance matrices

Fuzzy Clustering (1)

- Fuzzy Clustering
  - Let each cluster be represented by a parameter vector $\boldsymbol{\theta}_j$
  - To minimize the cost function
    - $J_q(\boldsymbol{\theta}, \mathbf{u}) = \sum_{i=1}^{N}\sum_{j=1}^{K} u_j^q(\mathbf{x}_i) d(\mathbf{x}_i, \boldsymbol{\theta}_j)$
      - Subject to
        - » $\sum_{j=1}^{K} u_j(\mathbf{x}_i) = 1,\ i = 1, \dots, N$
        - » $u_j(\mathbf{x}_i) \in [0,1]$
    - If $q = 1$
      - J_q = sum-of-squared error criterion
    - If $q > 1$
      - The criterion allows each pattern to belong to multiple clusters
  - The Lagrangian
    - $L = \sum_{i=1}^{N}\sum_{j=1}^{K} u_j^q(\mathbf{x}_i) d(\mathbf{x}_i, \boldsymbol{\theta}_j) - \sum_{i=1}^{N} \lambda_i (\sum_{j=1}^{K} u_j(\mathbf{x}_i) - 1)$

Fuzzy Clustering (2)

- Minimization of the criterion
  - $\frac{\partial}{\partial u_j(\mathbf{x}_i)} L = q u_j^{q-1}(\mathbf{x}_i) d(\mathbf{x}_i, \boldsymbol{\theta}_j) - \lambda_i = 0$

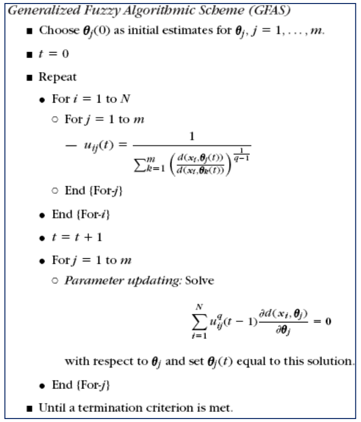
    $$\Rightarrow u_j(\mathbf{x}_i) = \left(\frac{\lambda_i}{q d(\mathbf{x}_i, \boldsymbol{\theta}_j)}\right)^{\frac{1}{q-1}}$$

    $$= \frac{1}{\sum_{s=1}^{K}\left(\frac{d(\mathbf{x}_i, \boldsymbol{\theta}_j)}{d(\mathbf{x}_i, \boldsymbol{\theta}_s)}\right)^{\frac{1}{q-1}}}$$

    - $\because \sum_{j=1}^{K} u_j(\mathbf{x}_i) = 1$
  - $\frac{\partial}{\partial \boldsymbol{\theta}_j} L = 0$

    $$\Rightarrow \sum_{i=1}^{N} u_j^q(\mathbf{x}_i) \frac{\partial d(\mathbf{x}_i, \boldsymbol{\theta}_j)}{\partial \boldsymbol{\theta}_j} = 0$$

P.715 [Theodoridis 09]



Generalized Fuzzy Algorithmic Scheme (GFAS)

```
■ Choose $\boldsymbol{\theta}_j(0)$ as initial estimates for $\boldsymbol{\theta}_j$, $j = 1, \ldots, m$.
■ $t = 0$
■ Repeat
    • For $i = 1$ to $N$
        ○ For $j = 1$ to $m$
            — $u_{ij}(t) = \frac{1}{\sum_{k=1}^{m}\left(\frac{d(x_i, \boldsymbol{\theta}_j(t))}{d(x_i, \boldsymbol{\theta}_k(t))}\right)^{\frac{1}{q-1}}}$
        ○ End {For-$j$}
    • End {For-$i$}
    • $t = t + 1$
    • For $j = 1$ to $m$
        ○ Parameter updating: Solve
              $\sum_{i=1}^{N} u_{ij}^q(t-1) \frac{\partial d(x_i, \boldsymbol{\theta}_j)}{\partial \boldsymbol{\theta}_j} = 0$
          with respect to $\boldsymbol{\theta}_j$ and set $\boldsymbol{\theta}_j(t)$ equal to this solution.
    • End {For-$j$}
■ Until a termination criterion is met.
```

Fuzzy Clustering (3)

- Point representative
  - Let each cluster be represented by a vector $\boldsymbol{\mu}_j$
- Fuzzy K-means
  - The dissimilarity
    - $d(\mathbf{x}_i, \boldsymbol{\mu}_j) = (\mathbf{x}_i - \boldsymbol{\mu}_j)^T \mathbf{A} (\mathbf{x}_i - \boldsymbol{\mu}_j)$
      - $\mathbf{A}$ is a symmetric, positive definite matrix
  - $\frac{\partial L}{\partial \boldsymbol{\theta}_j} = \frac{\partial L}{\partial \boldsymbol{\mu}_j} = 0$

    $$\Rightarrow \sum_{i=1}^{N} u_j^q(\mathbf{x}_i) \frac{\partial d(\mathbf{x}_i, \boldsymbol{\mu}_j)}{\partial \boldsymbol{\mu}_j} = \sum_{i=1}^{N} u_j^q(\mathbf{x}_i)\, 2\mathbf{A}(\mathbf{x}_i - \boldsymbol{\mu}_j) = 0$$

    $$\Rightarrow \boldsymbol{\mu}_j = \frac{\sum_{i=1}^{N} u_j^q(\mathbf{x}_i)\mathbf{x}_i}{\sum_{i=1}^{N} u_j^q(\mathbf{x}_i)}$$

Fuzzy Clustering (4)

- Example 14.5.a (p.706, [Theodoridis 09])
  - Consider the data used in Example 14.1
  - Let $q = 2$ and assign $\mathbf{x} \rightarrow C_i$ if $i = \underset{j=1,\dots,K}{\operatorname{argmax}} u_j^q(\mathbf{x})$

(a) $\boldsymbol{\mu}_1 = \begin{bmatrix}1\\1\end{bmatrix}$ $\boldsymbol{\mu}_2 = \begin{bmatrix}3.5\\3.5\end{bmatrix}$ $\boldsymbol{\mu}_3 = \begin{bmatrix}6\\1\end{bmatrix}$ (b) $\boldsymbol{\mu}_1 = \begin{bmatrix}1\\1\end{bmatrix}$ $\boldsymbol{\mu}_2 = \begin{bmatrix}2\\2\end{bmatrix}$ $\boldsymbol{\mu}_3 = \begin{bmatrix}3\\1\end{bmatrix}$

$\hat{\boldsymbol{\mu}}_1 = \begin{bmatrix}1.37\\0.71\end{bmatrix}$ $\hat{\boldsymbol{\mu}}_2 = \begin{bmatrix}3.14\\3.12\end{bmatrix}$ $\hat{\boldsymbol{\mu}}_3 = \begin{bmatrix}5.08\\1.21\end{bmatrix}$ $\hat{\boldsymbol{\mu}}_1 = \begin{bmatrix}1.6\\0.12\end{bmatrix}$ $\hat{\boldsymbol{\mu}}_2 = \begin{bmatrix}1.15\\1.67\end{bmatrix}$ $\hat{\boldsymbol{\mu}}_3 = \begin{bmatrix}3.37\\2.1\end{bmatrix}$

	C1	*C2*	*C3*
1st distribution	98	2	0
2nd distribution	14	84	2
3rd distribution	11	0	89

	C1	*C2*	*C3*
1st distribution	51	46	3
2nd distribution	14	47	39
3rd distribution	43	0	57

Spectral Clustering (1)

- Spectral clustering
  - A graph-based technique to unravel the structural properties of a graph
    - Given $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}, \mathbf{x}_i \in R^d$
    - Bi-partition the set into 2 clusters
    - The cluster indicator for $\mathbf{x}_i$ is $y_i \in \{-1,1\}$
  - Constructing a graph $G = (V, E)$
    - Each node corresponds to a point $\mathbf{x}_i$
    - Two vertices are connected with an edge if $\|\mathbf{x}_i - \mathbf{x}_j\|^2 < \varepsilon$
  - Weighting each edge
    - $W(i,j) = \begin{cases} \frac{\exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)}{\sigma^2}, & \text{if } \|\mathbf{x}_i - \mathbf{x}_j\|^2 < \varepsilon \\ 0, & \text{otherwise} \end{cases}$

Spectral Clustering (2)

- Choosing an appropriate clustering criterion
  - Cut
    - $cut(A,B) = \sum_{i\in A, j\in B} W(i,j)$
      - A and B are the resulting clusters
    - Selecting A and B so that $cut(A,B)$ is minimized
      - Set of edges connecting A and B have minimum sum of weights
    - However, minimum cut criterion would results in clusters of small size of isolated points (least similar with the rest of the nodes)

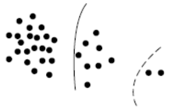


Fig. 15.8 [Theodoridis et. al.]

FIGURE 15.8

The cut criterion has the tendency to form small clusters of isolated points, as for example the two points separated by the dotted line. A more natural clustering for this case results by the full line.

Spectral Clustering (3)

- Choosing an appropriate clustering criterion (cont.)
  - Normalized cut
    - To minimize the cut and also to keep the sizes of the clusters large
    - $Ncut(A,B) = \frac{cut(A,B)}{Vol(A)} + \frac{cut(A,B)}{Vol(B)} = cut(A,B)\left(\frac{1}{Vol(A)} + \frac{1}{Vol(B)}\right)$
      - Where the volume or the degree of A measures the importance of the vertices in A relative to other vertices
        - » $Vol(A) = \sum_{i\in A, j\in V} W(i,j)$
    - A small and isolated cluster has a small volume and thus will result in large $Ncut$
    - Minimization of $Ncut$ is an NP-hard task
    - However, if allowing relaxation of the indicator y_i to real values, the problem reduces to minimizing the Laplacian of the graph
      - An approximate solution

Spectral Clustering (4)

- The relaxed problem
  - Let the cluster indicator for $\mathbf{x}_i$ be
    - $y_i = \begin{cases} \frac{1}{Vol(A)}, \text{if } i \in A \\ -\frac{1}{Vol(B)}, \text{if } i \in B \end{cases}$
  - Define the diagonal weight matrix $\mathbf{D}$
    - Measuring the significance of a node
    - $D_{ii} = \sum_{j \in V} W(i,j)$
      - $Vol(A) = \sum_{i \in A} D_{ii} = \sum_{i \in A, j \in V} W(i,j)$
  - Define the Laplacian matrix $\mathbf{L}$
    - $\mathbf{L} \equiv \mathbf{D} - \mathbf{W}$
    - Symmetric and positive semidefinite

Spectral Clustering (5)

- The relaxed problem (cont.)
  - We have
    - $\mathbf{y}^T\mathbf{L}\mathbf{y} = \sum_{i \in V} \sum_{j \in V} (y_i - y_j)^2 W(i,j)$

      $= \sum_{i \in A} \sum_{j \in B} \left(\frac{1}{Vol(A)} + \frac{1}{Vol(B)}\right)^2 cut(A,B) \propto \left(\frac{1}{Vol(A)} + \frac{1}{Vol(B)}\right)^2 cut(A,B)$
    - $\mathbf{y}^T\mathbf{D}\mathbf{y} = \sum_{i \in A} y_i^2 D_{ii} + \sum_{j \in B} y_j^2 D_{jj} = \frac{1}{Vol^2(A)} Vol(A) + \frac{1}{Vol^2(B)} Vol(B)$

      $= \frac{1}{Vol(A)} + \frac{1}{Vol(B)}$
  - Then, minimizing $Ncut(A,B)$ is equivalent with minimizing
    - $J = \frac{\mathbf{y}^T\mathbf{L}\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}} \propto Ncut(A,B)$
    - Note that
      - $\mathbf{y}^T\mathbf{D}\mathbf{1} = 0$
        - » $\mathbf{1}$ is a column vector of ones

Spectral Clustering (6)

- The relaxed problem (cont.)
  - To minimize $J = \frac{\mathbf{y}^T\mathbf{L}\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}}$
    - Under the condition that $\mathbf{y}^T\mathbf{D}\mathbf{1} = 0$
  - Let $\mathbf{z} = \mathbf{D}^{\frac{1}{2}}\mathbf{y}$
  - Then
    - $J = \frac{\mathbf{y}^T\mathbf{L}\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}} = \frac{\mathbf{z}^T\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{1/2}\mathbf{z}}{\mathbf{z}^T\mathbf{z}} = \frac{\mathbf{z}^T\tilde{\mathbf{L}}\mathbf{z}}{\mathbf{z}^T\mathbf{z}}$
    - The constraint becomes $\mathbf{z}^T\mathbf{D}^{1/2}\mathbf{1} = 0$
    - $\tilde{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{1/2}$ is the normalized graph Lapalcian matrix
    - $\mathbf{D}^{1/2}\mathbf{1}$ is an eigenvector corresponding to a zero eigenvalue
    - Thus the minimization is achieved
      - when $\mathbf{z}$ is the eigenvector corresponding to the second smallest eigenvalue of $\tilde{\mathbf{L}}$

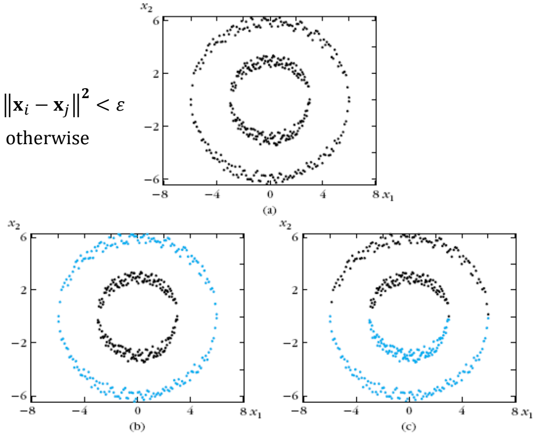
Spectral Clustering (7)

- The relaxed problem (cont.)
  - Step 1
    - Construct a graph $G = (V, E)$ and form the proximity matrix $\mathbf{W}$
  - Step 2
    - Form the matrices
      - $D_{ii} = \sum_{j \in V} W(i,j)$, $\mathbf{L} = \mathbf{D} - \mathbf{W}$, $\tilde{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{1/2}$
  - Step 3
    - Perform the eigenanalysis $\tilde{\mathbf{L}}\mathbf{z} = \lambda\mathbf{z}$
    - Compute the eigenvetor $\mathbf{z}_1$ corresponding to the 2nd smallest eigenvalue
    - Compute the vector $\mathbf{y} = \mathbf{D}^{-\frac{1}{2}}\mathbf{z}_1$
  - Step 4
    - Discretize the components of $\mathbf{y}$ according to a threshold value

Spectral Clustering (8)

$$W(i,j) = \begin{cases} \frac{\exp\left(-\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right)}{\sigma^2}, & \text{if } \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2 < \varepsilon \\ 0, & \text{otherwise} \end{cases}$$

$$\sigma^2 = 2$$
$$\varepsilon = 2$$



[Theodoridis et. al.]

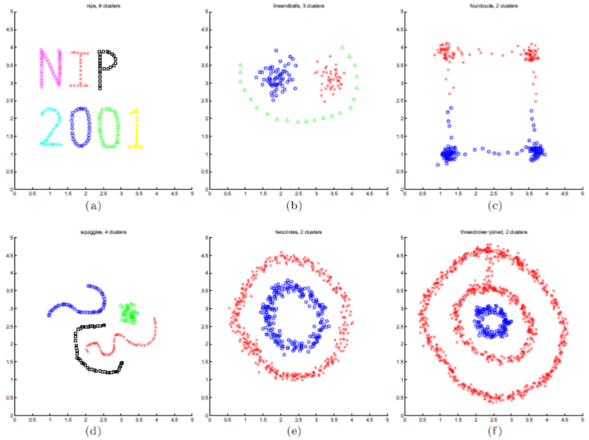
FIGURE 15.9

(a) The data set. (b) The two clusters (denoted by different colors) obtained by the spectral clustering algorithm. (c) The two clusters obtained by the *k*-means algorithm.

Spectral Clustering (9)

- To directly find K clusters [Ng et al, 2001]
  - Steps 1 & 2
  - Step 3
    - Perform the eigenanalysis $\tilde{\mathbf{L}}\mathbf{z} = \lambda\mathbf{z}$
    - Find the smallest K eigenvectors $\{\mathbf{z}_1, \ldots, \mathbf{z}_K\}$
    - Form the matrix by stacking the eigenvectors in columns
      - $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_K] \in R^{N\times K}$
    - Form the matrix $\mathbf{T}$ from $\mathbf{Z}$ by normalizing each row to be unit length
      - $t_{ij} = z_{ij} / \left(\sum_k z_{ik}^2\right)^{\frac{1}{2}}$
  - Step 4
    - Treat each row of $\mathbf{T}$ as a point in R^K
    - Cluster them using k-means
  - Step 5
    - Assign $\mathbf{x}_i$ to cluster k if row i of $\mathbf{T}$ was assigned to cluster k

Spectral Clustering (10)



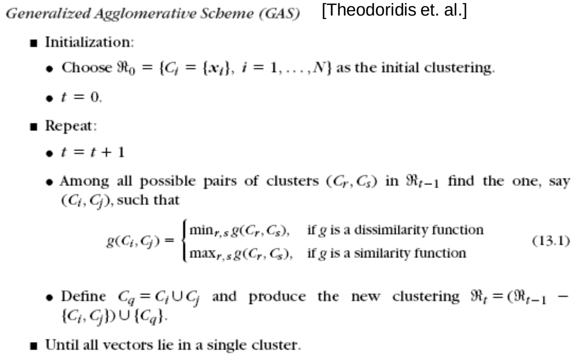
A. Ng, M. Jordan, and Y. Weiss, “On Spectral Clustering: Analysis and an algorithm,” In *NIPS,* 2001.

Hierarchical Algorithms (1)

- Two main approaches
  - Agglomerative algorithm (bottom-up procedure)
    - Produce a sequence of clusterings of decreasing K at each step
      - No way to recover from a poor clustering in an earlier level
  - Divisive clustering (top-down procedure)
    - Produce a sequence of clusterings of increasing K at each step
- Both approaches are just heuristics
  - Do not optimize any objective function
  - Difficult to assess the quality of clustering

Hierarchical Algorithms (2)

- Agglomerative algorithm (bottom-up procedures)
  - Start with N singleton clusters
  - Successively merge two nearest clusters



Generalized Agglomerative Scheme (GAS) [Theodoridis et. al.]

```
■ Initialization:
    • Choose $\Re_0 = \{C_i = \{x_i\},\ i = 1, \ldots, N\}$ as the initial clustering.
    • $t = 0$.
■ Repeat:
    • $t = t + 1$
    • Among all possible pairs of clusters $(C_r, C_s)$ in $\Re_{t-1}$ find the one, say
      $(C_i, C_j)$, such that
          $g(C_i, C_j) = \begin{cases} \min_{r,s} g(C_r, C_s), & \text{if } g \text{ is a dissimilarity function} \\ \max_{r,s} g(C_r, C_s), & \text{if } g \text{ is a similarity function} \end{cases}$   (13.1)
    • Define $C_q = C_i \cup C_j$ and produce the new clustering $\Re_t = (\Re_{t-1} - \{C_i, C_j\}) \cup \{C_q\}$.
■ Until all vectors lie in a single cluster.
```

Hierarchical Algorithms (3)

- Agglomerative algorithm (cont.)
  - $d(C_i, C_j)$: the dissimilarity between clusters
  - Single link (nearest neighbor)
    - The distance is that of the two closest members of each group
      - $d_{SL}(C_i, C_j) = \min_{\mathbf{x} \in C_i, \mathbf{x}' \in C_j} d(\mathbf{x}, \mathbf{x}')$
    - Can produce clusters with large diameters
  - Complete link (furthest neighbor)
    - The distance is that of the two most distant pairs
      - $d_{CL}(C_i, C_j) = \max_{\mathbf{x} \in C_i, \mathbf{x}' \in C_j} d(\mathbf{x}, \mathbf{x}')$
    - Tend to produce compact clusters
  - Average link
    - $d_{AVG}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{\mathbf{x} \in C_i} \sum_{,\mathbf{x}' \in C_j} d(\mathbf{x}, \mathbf{x}')$

Hierarchical Algorithms (4)

- Example (p.656, [Theodoridis et. al.])
  - The pattern matrix and the dissimilarity matrix

$$D(X)=\begin{bmatrix}\mathbf{x}_1^T\\ \mathbf{x}_2^T\\ \mathbf{x}_3^T\\ \mathbf{x}_4^T\\ \mathbf{x}_5^T\end{bmatrix}=\begin{bmatrix}1 & 1\\ 2 & 1\\ 5 & 4\\ 6 & 5\\ 6.5 & 6\end{bmatrix} \quad P(X)=\begin{bmatrix}0 & 1 & 5 & 6.4 & 7.4\\ 1 & 0 & 4.2 & 5.7 & 6.7\\ 5 & 4.2 & 0 & 1.4 & 2.5\\ 6.4 & 5.7 & 1.4 & 0 & 1.1\\ 7.4 & 6.7 & 2.5 & 1.1 & 0\end{bmatrix}$$

The merging process can be represented by a binary tree (called a dendrogram)

The height of the braches represents the dissimilarity

Cutting the dendrogram at a level results in a clustering

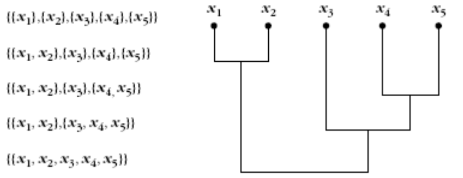
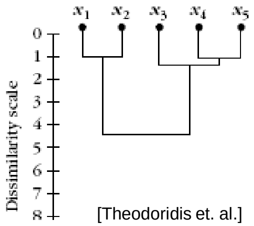

$\{\{x_1\},\{x_2\},\{x_3\},\{x_4\},\{x_5\}\}$
$\{\{x_1, x_2\},\{x_3\},\{x_4\},\{x_5\}\}$
$\{\{x_1, x_2\},\{x_3\},\{x_4, x_5\}\}$
$\{\{x_1, x_2\},\{x_3, x_4, x_5\}\}$
$\{\{x_1, x_2, x_3, x_4, x_5\}\}$
x_1
x_2
x_3
x_4
x_5

FIGURE 13.1

The clustering hierarchy for X of Example 13.1 and its corresponding dendrogram.



[Theodoridis et. al.]

Fig. 13.2 (b): single ink clustering

Hierarchical Algorithms (5)

- Divisive algorithm (top-down procedures)
  - Start with one cluster with N samples
    - Among all possible pairs of sub-clusters that form a partition of clusters at current stage, find the pair that optimize the clustering criterion

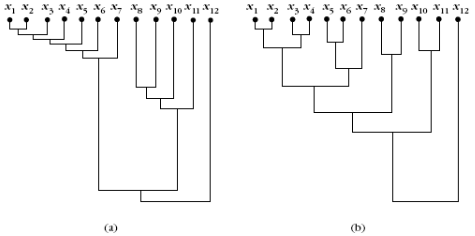
Generalized Divisive Scheme (GDS)

[Theodoridis et. al.]

```
■ Initialization
    • Choose $\Re_0 = \{X\}$ as the initial clustering.
    • $t = 0$
■ Repeat
    • $t = t + 1$
    • For $i = 1$ to $t$
        ○ Among all possible pairs of clusters $(C_r, C_s)$ that form a partition of $C_{t-1,i}$,
          find the pair $(C^1_{t-1,i}, C^2_{t-1,i})$ that gives the maximum value for $g$.
    • Next $i$
    • From the $t$ pairs defined in the previous step choose the one that maximizes
      $g$. Suppose that this is $(C^1_{t-1,j}, C^2_{t-1,j})$.
    • The new clustering is
          $\Re_t = (\Re_{t-1} - \{C_{t-1,j}\}) \cup \{C^1_{t-1,j}, C^2_{t-1,j}\}$
    • Relabel the clusters of $\Re_t$.
■ Until each vector lies in a single distinct cluster.
```

Hierarchical Algorithms (6)

- Determining the number of clusters
  - To search in the dendrogram for clusters that have a large lifetime
    - The absolute value of the difference between the proximity level at which it is created and at which it is absorbed into a larger cluster


x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12}
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12}
(a)
(b)

[Theodoridis et. al.]

FIGURE 13.17

(a) A dendrogram that suggests that there are two major clusters in the data set. (b) A dendrogram indicating that there is a single major cluster in the data set.

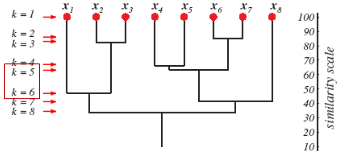


Fig. 10.11 [Duda 01]

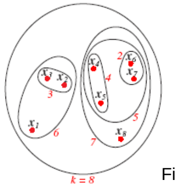


Fig. 10.12 [Duda 01]

Cluster Validity (1)

- Given a set of clusterings
  - e.g. a set of parameters, such as the number of clusters, and the initial estimate of the parameter vectors
- Goal
  - To choose the best one according to a prespecified criterion

- Case 1 (K is not one of the parameters)
  - Run the cluster algorithm for a wide range of values of its parameters
  - Choose the widest range for which K remains constant
  - Choose the parameter that corresponds to the middle of this range

Cluster Validity (2)

- Example (p. 887 [Theodoridis et. al.])
  - The data set X consists of 3 groups of 100 2-D vectors
  - (a) K remains constant for the parameter r between 37 and 67
    - Choosing r = 52 and K = 3
  - (b) K remains constant for r between 7 and 46
    - Choosing r = 26 and K = 1

[Theodoridis et. al.]

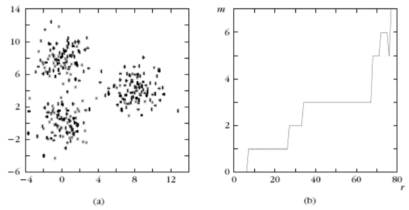


FIGURE 16.2

(a) Three well-separated clusters. (b) The plot of the number of clusters m versus the resolution parameter r, using the binary morphology clustering algorithm (BMCA).

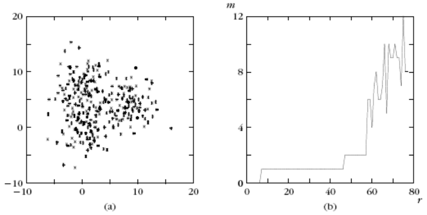


FIGURE 16.3

(a) Three overlapped clusters. (b) The plot of m versus r.

Cluster Validity (3)

- Case 2 (K is one of the parameters)
  - Run the cluster algorithm for values of K between K_{min} and K_{max}
  - For each K
    - Run the algorithm r times using different sets of parameters
  - Plot the performance index q versus K
    - e.g., reconstruction error, within-cluster dissimilarity
    - q generally decreases with increasing K
  - Search for K at which a significant local change (knee) in q occurs
    - When $K <$ ideal K^*
      - q tends to decrease substantially with increasing K
    - When$K >$ ideal K^*
      - q tends to decrease smaller with increasing K

Cluster Validity (4)

- Example (p. 879 [Theodoridis et. al.])
  - The data set consists of 4 compact and well-separated clusters
    - N: number of data in a data set
    - l: dimension
  - The higher the dimensionality, the sharper the knee at $K = 4$
  - The larger of the data size N, the knee at $K = 4$ becomes sharper

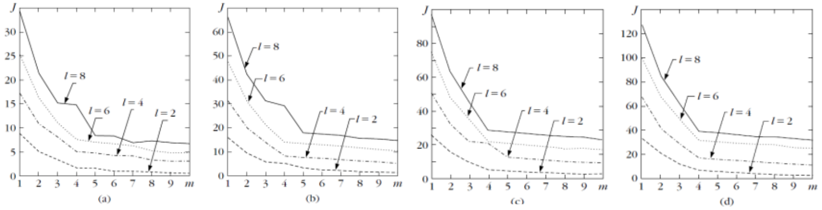


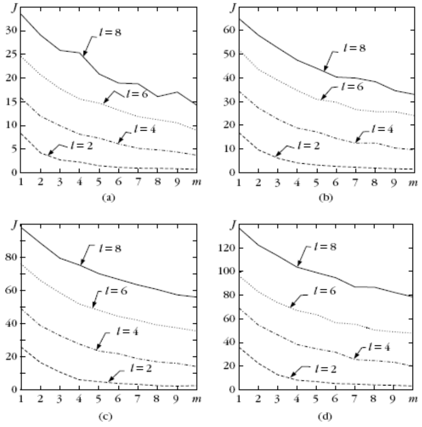
FIGURE 16.4

Plots of J versus m for (a) $N = 50$, (b) $N = 100$, (c) $N = 150$, (d) $N = 200$, for clustered data.

[Theodoridis et. al.]

Cluster Validity (5)

- Example (cont.)
  - The data set is randomly generated without clustering structure
  - There are no sharp knees in the plots



[Theodoridis et. al.]

FIGURE 16.5

Plots of J versus m for (a) $N = 50$, (b) $N = 100$, (c) $N = 150$, (d) $N = 200$, for random data.